LArSoft - Bug #24985

Some missing size checks in TrajClusterAlg::FindJunkTraj

09/17/2020 06:36 PM - Kenneth Herner

Status:	Closed	**Start date:**	09/17/2020
Priority:	Normal	**Due date:**	
Assignee:	Bruce Baller	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Occurs In:		**Co-Assignees:**	
Experiment:	DUNE		

Description

Hi everyone,

We recently hit a problem in a DUNE MC request where we were seeing segfault at a ~40% rate in reco (running the standard reco fcl file, so nothing "custom" is happening). I traced it to several calls in the FindJunkTraj function in TrajClusterAlg.cxx. What seems to be happening is that some members of TCSlice are getting filled with bogus values, and then when trying to access various vector elements using these values as an index (i.e. the value exceeds the size of the vector) you get segfaults because of invalid references. I found four cases where this could happen though there could be more. This is all based on larreco v08_32_10.

1) Line 691 of TrajClusterAlg.cxx:

for(unsigned int iwire = slc.firstWire[plane]; iwire < slc.lastWire[plane] - 3; ++iwire)

In one of the files that showed a segfault we observed that slc.lastWire[plane] returned a bogus value (2^32-1 actually) and then you have problems when trying to do 693-699 because you're effectively trying to get slc.wireHitRange[plane][2^32-1] which of course work for any value in the second index that is >= the size of slc.wireHitRange[plane]. With a check in place to stop the loop once iwire gets to the size of the vector -1, it works.

2) Lines 700-702:

```
for(unsigned int iht = ifirsthit; iht <= ilasthit; ++iht) {
      auto& islHit = slc.slHits[iht];
        if(islHit.InTraj != 0) continue;
```

One can get a segfault on 702 if iht exceeds slc.slHits.size()-1 because you'll get an invalid reference on 701.

```
3) 708-710:
```

```
for(unsigned int jht = jfirsthit; jht <= jlasthit; ++jht) {
          auto& jslHit = slc.slHits[jht];
          if(jslHit.InTraj != 0) continue;
```

Same thing as 2. Note that ifirsthit, ilasthit, jfirsthit, and jlasthit are set in 696-699.

4) Lines 730-735:

```
for(unsigned int kwire = loWire; kwire <= hiWire; ++kwire) {
      if(slc.wireHitRange[plane][kwire].first == UINT_MAX) continue;
        unsigned int kfirsthit = slc.wireHitRange[plane][kwire].first;
        unsigned int klasthit = slc.wireHitRange[plane][kwire].second;
        for(unsigned int kht = kfirsthit; kht <= klasthit; ++kht) {
          if(slc.slHits[kht].InTraj != 0) continue;
```

Same idea if kwire and/or kht gets too big; there's no check that they're within size of the respective vectors.

To be sure, this is a rare problem and probably points to other things going wrong such that we should not trust the event, but it's often enough to notice and cause problems. There are of course many ways around the problems ranging from just skipping the wire(s)/planes in question to bailing on the whole function.

Thanks for your attention,

Ken

History

#1 - 09/17/2020 11:48 PM - Lynn Garren

Bruce, is this in your code?

#2 - 09/18/2020 10:57 AM - Lynn Garren

- Assignee set to Bruce Baller

- Status changed from New to Feedback

There hasn’t been a lot of bounds checking in FindJunkTraj because this algorithm is called late in the chain so your comment about bogus values in TCSlice is disturbing. Can you please point me to a data file?

BTW, one can turn off this algorithm using the SkipAlgs fcl vector like this:

physics.producers.trajcluster.TrajClusterAlg.SkipAlgs: [“JunkTj”, “<SkippedAlgorithm1>”, “<SkippedAlgorithm2>”, …]

What version of dunetpc is being used? I would like to peruse the fcl job files. I see that some SkipAlgs definitions are out of date in dunetpc v09_01_00.

#3 - 09/18/2020 11:43 AM - Kenneth Herner

Hi Bruce,

This is all in dunetpc v08_62_00. I've moved a couple of example files over if you want to look them over:

/dune/data/users/kherner/data_files_24985/NNBarAtm_hA_LFG_dune10kt_1x2x6_36716361_11_gen_g4_detsim.root # the 9th event, 4409, is the one that segfaults

/dune/data/users/kherner/data_files_24985/NNBarAtm_hA_LFG_dune10kt_1x2x6_36716361_36_gen_g4_detsim.root # the 136th event, 14536, is the one that segfaults

After playing around some more, if I change line 691 to be

for(unsigned int iwire = slc.firstWire[plane]; iwire < slc.lastWire[plane] - 3 && iwire < slc.wireHitRange[plane].size(); ++iwire)

That seems to be enough to avoid any weirdness. I wonder if something upstream uses 0xffffffff as an initial value (expecting it to be -1 perhaps) and/or an indication of some error for slc.firstWire[plane] and slc.lastWire[plane], but then it gets interpreted as an unsigned int here. That's just a guess though.

#4 - 09/18/2020 11:44 AM - Kenneth Herner

And the fcl is just standard_reco_dune10kt_1x2x6.fcl, which in turn calls standard_reco_dune10kt.fcl.

#5 - 09/19/2020 01:20 PM - Bruce Baller

- % Done changed from 0 to 30

- File evd.twq-proj.36716361.4409.pdf added

Kenneth,

I understand why the seg fault occurred. These simulated NNBar events have a unique pattern of a small number of hits on a few wires. The code that determines the range of wires to consider in each TPC and plane obviously didn't handle this situation correctly. The fix that you made locally looks fine. I applied a slightly different approach to bounds checking in a v09_01_00 larreco feature branch. The attached event display image shows the TrajCluster cluster reconstruction of event 4409 using this feature branch.

I don't know how to back port this change to larreco v08_32_10, if that is even possible. It appears that you aren't involved in a MC production campaign so maybe you are OK using your local build. Let me know if that is not the case, otherwise I think this issue can be closed.

BTW, the version of TrajCluster on my feature branch has improved 2D cluster reconstruction and also produces 3D tracks, obviating the need to use PMA.

Bruce

#6 - 09/21/2020 10:55 AM - Lynn Garren

Bruce, will you be making a PR soon for the head of larreco?

#7 - 09/21/2020 11:01 AM - Bruce Baller

Lynn,

The feature branch has significant changes. I hope to submit a PR by the end of this week. If that is unsatisfactory, I could submit a separate PR for the bug fix.

Bruce

#8 - 09/21/2020 11:06 AM - Kenneth Herner

Hi Bruce,

This actually showed up in tests for a fairly large production run for the NDK group that we wanted to start soon with v08_62_00. However they're playing around with refactored fcl files for for detector MC now, so they may ask to switch to a v9 release for the final version. As for the v8_xx_xx larreco branch, if all else fails maybe it's enough to backport my local fix (just in case they don't want to move to v9); that should avoid most of the problematic stuff anyway.

Regards,
Ken

#9 - 09/21/2020 11:18 AM - Lynn Garren

Bruce, if it's not too much work, it would be helpful to get a bug fix PR followed by your other updates. This allows us to understand the changes better.

#10 - 09/21/2020 11:19 AM - Bruce Baller

I agree. I am working on it now.

#11 - 09/23/2020 09:28 AM - Bruce Baller

- % Done changed from 30 to 90

#12 - 09/24/2020 03:09 PM - Lynn Garren

- % Done changed from 90 to 100

- Status changed from Feedback to Resolved

Bruce supplied larreco PR 20. This is part of larsoft v09_04_01 (and v09_04_00).

#13 - 09/28/2020 10:35 AM - Kyle Knoepfel

- Status changed from Resolved to Closed

Files

evd.twq-proj.36716361.4409.pdf	25.8 KB	09/19/2020	Bruce Baller